Docket No.: SON-2047
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Akihiko Koh et al.

Application No.: 09/802,857

Confirmation No.: 3304

Filed: March 12, 2001

Art Unit: 2192

For: DATA PROCESSING APPARATUS PERFORMING PREDETERMINED DATA PROCESSING IN ACCORDANCE WITH INSTRUCTION CODES READ FROM A PROGRAM MEMORY STORING A PROGRAM

Examiner: M. J. Yigdall

AMENDMENT IN RESPONSE TO NON-FINAL OFFICE ACTION

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

INTRODUCTORY COMMENTS

In response to the Office Action dated September 10, 2007, please amend the above-identified U.S. patent application as follows:

Amendments to the Specification begin on page 2 of this paper.

Amendments to the Claims are reflected in the listing of claims which begins on page 3 of this paper.

Remarks/Arguments begin on page 7 of this paper.

AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph beginning at page 21, line 8, with the following rewritten paragraph.

-- In the interrupt processing, the CPU 10 reads the instruction codes sequentially from the memory address designated by the program address output to the address bus and executes the same (step S6). Namely, as shown in FIG. 5, the debugged program stored in the area from the memory address F000H in the RAM 50 is read sequentially and executed. At the end of the program, an instruction code showing the return address for after the interrupt processing (for example, in the example of FIG. 5, "RET BADR1+1") is stored. Upon reading this instruction code, the CPU 10 finishes the interrupt processing, then sets the return address "BADR1+1" in the program counter (step S7). --

Please replace the paragraph beginning at page 21, line 21, with the following rewritten paragraph.

-- As shown in FIG. 5, "BADR1" is the end address of the buggy part of the program stored in the ROM 30. For this reason, by setting "BADR1+1" in the CPU 10 as the return address, after the interrupt processing, the CPU 10 reads the program codes from the code next to the buggy part of the program to continue the processing. --